



**EMBEDDED
LINUX
CONFERENCE**

@



THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
EUROPE

Embedded Linux Ecosystem

Birds of a Feather Session

Tim Bird – Principal Software Engineer, Sony

#ossummit @twitterhandle



Abstract

In the Birds-of-a-Feather session, Tim will discuss the current status of the embedded Linux ecosystem. Who are the major players? Who is doing a good job of contributing upstream? Are there things that multiple people are working on, where they could combine or share their efforts? What things can we do to make sure that new developers feel welcome? What can we do to make it so that developing, distributing and maintaining an embedded Linux product is as easy as possible? What resources (e.g. the elinux wiki, or the embedded-linux kernel mailing list) could we utilize more effectively, to build our ecosystem?

Tim doesn't have all the answers for these. But come join us for an open discussion of the issues, and let's brainstorm together ideas to make our embedded Linux ecosystem better.

Agenda

- Status
- Overview of Issues
- Discussion

Disclaimer: I apologize in advance for not listing your company, organization, project, market, issue, etc.

Status

Status of Embedded Linux

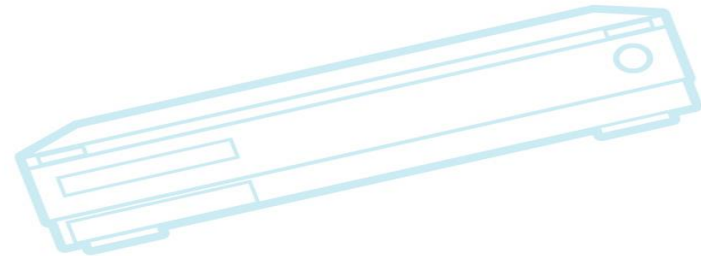
- In June, I gave my “Status of Embedded Linux Talk”
 - I won’t repeat that here
 - See https://elinux.org/ELC_2022_Presentations
- Let’s quickly review the scorecards...



CE Workgroup

Technology Scorecard

- Based on kernel contributions in the last few years:
- System size – done
- Boot time – done
- Power management – done
- Realtime – done
- Security – in progress
- Audio/Video/Graphics – in progress





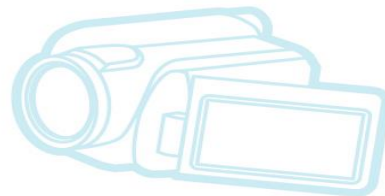
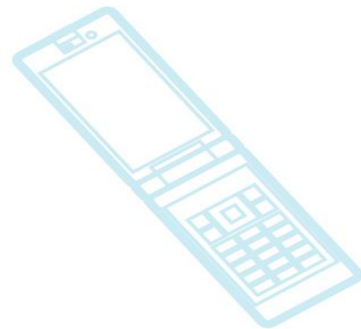
CE Workgroup

Real Technology Scorecard

- System size – done
- Boot time – done
- Power management – done??
- Realtime – done??
- Security – in progress

Recognise now that all of these are “holistic”.

- All of these require pervasive, constant maintenance, distributed throughout the code base

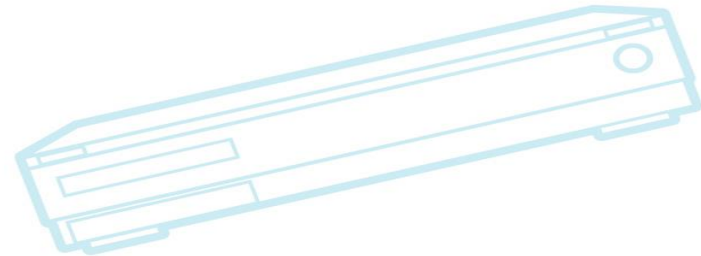
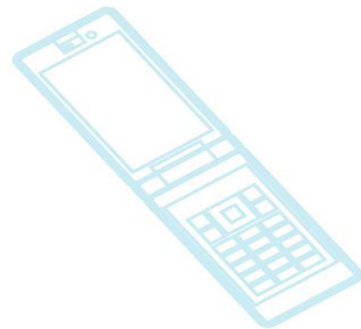




CE Workgroup

Development Scorecard

- Build systems/Distros – good
- Documentation – good?
- Training/Consulting - good
- Toolchains – good
- Debugging capabilities – good
- Languages – good
- Test Systems – in progress
- Hardware support – in progress

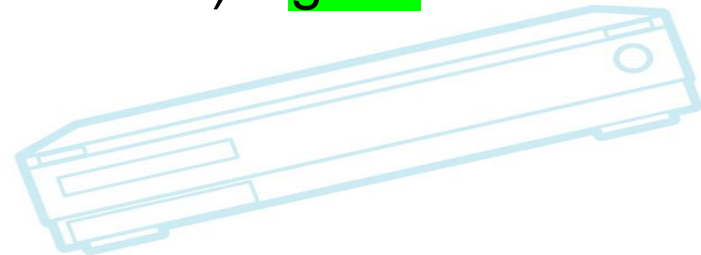
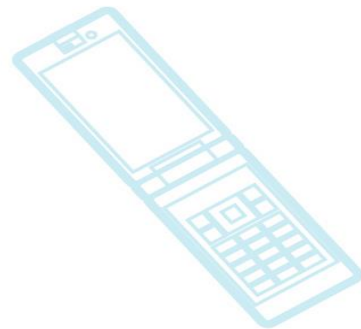




CE Workgroup

Markets Scorecard

- Drones - good
- Robots – good
- Cars – good
- Space systems – improving
- Routers - good
- Consumer electronics (TVs, DVRs, Cameras) - good
- Mobile Phones – good



Issues

First – two observations...

Paradox of embedded Open Source

- Open source effects come from sharing and collaborating
- The essence of embedded is customization
 - In embedded, everyone is doing something unique and custom

Who are the ecosystem members?

- Silicon vendors, makers, associations
 - ARM, RISC-V
 - Intel, Qualcomm, MediaTek, Broadcom, NVidia, Texas Instruments, NXP, Samsung, etc.
 - Linaro, CIP
- Hardware vendors
 - IP block vendors, peripheral vendors
- Embedded Linux Vendors
 - Baylibre, Bootlin, Collabora, Konsulko, Linaro, Linutronix, Konsulko, Mentor, Pengutronix, Toradex
- Board vendors
 - Raspberry Pi, Beaglebone, etc.
- Product manufacturers
 - Consumer Electronic product manufacturers (phone, TV, cameras, media players)
 - Automotive, Robotics, Routers, many others (there's a long tail) - tractors, rockets
- Industrial users
- Makers/Hobbyists
- Distribution and build system vendors and projects
 - Debian, Ubuntu, Yocto Project, AGL, ROS, OpenWRT, OpenEmbedded, YoctoProject, Buildroot
- Upstream developers

- Upstreaming
- Testing
- Resources
- Information availability
- Connecting with each other
- Ecosystem growth

- How many lines of code does your embedded product have, that are out-of-mainline?
 - ie, that you have to develop and maintain

Who is doing a good job of contributing upstream?

- Scripts to analyze recent commits
 - using `check-embedded-company-stats.sh`

Company or Organization	commits (SINCE=2021)
BayLibre	161
Bootlin	400
Collabora	486
Konsulko	3L
Linaro	2407
Linutronix	543
Pengutronix	727
Toradex	128

More and better Collaboration

- What problems do we share, that we're working on in isolation?
 - (Or that no one is working on)?
- Problems with:
 - kernel?
 - distros?
 - testing?
 - compliance?
 - community infrastructure?
- Can we combine or share (or start) efforts?

Increasing our ecosystem

- Bringing more people to the community
 - Embedded developers exist, they just don't show up in upstream or at these events or in forums
 - Estimate is that there are between 250K and 1 million embedded Linux developers
- What things can we do to make sure that new developers feel welcome?
- Are there obstacles to participating?

- What can we do to make it so that
 - developing
 - distributing, and
 - maintaining

an embedded Linux product is as easy as possible?

Resources

- What resources could we utilize more effectively, to build our ecosystem?
- eLinux wiki
- Mailing lists
- Conferences
- Other resources

- <https://elinux.org/>
 - Does anyone here use this wiki?
- Status:
 - Hosted by Oregon State University Open Source Lab (OSUOSL)
 - Content: some areas active, some pretty stale
 - Good repository of past event slides and links to videos

Issue:

- Administrator will likely get defunded next year
 - Main job is MediaWiki upgrades and spam control (approval of user accounts, and rolling back spam content)
 - Can we raise funds for administration?
 - Should we move to a volunteer administrator?

Issue:

- Has lots of stale content
- Does anyone here contribute to this wiki?
- Does anyone know about the topical guide to past presentations? (good through 2019, I think)
- Are there any volunteers to help clean up the content?

Mailing Lists

- linux-embedded kernel mailing list
 - Exists, but is very seldom used
 - Is intended to be specific to the kernel
 - This discussion wouldn't be appropriate here
- celinux-dev mailing list
 - Exists, but is also seldom used
 - Was intended to discuss “consumer electronics Linux development” issues
- Should we get a new general-purpose mailing list?
 - What is the goal? Discuss high-level stuff like this.

- General events:
 - Embedded Linux Conference, Linaro Connect, Automotive Linux Summit, Yocto Dev. Days, Zephyr Summit, Embedded Recipes
- Regional conferences:
 - FOSDEM, Japan Jamboree, etc.

Feedback on ELC

- Should we split ELC from Open Source Summit to have a stronger focus on embedded?
- Frequency (currently twice a year)
 - Could it be once a year (e.g. alternating between Europe and US)?
 - Is that too far apart?
- Content and format:
 - More like Plumbers micro-conferences?
 - stronger themed tracks, curation, moderation, etc.
 - Work in progress (plumbers) vs work that exists, for different levels of expertise (ELC)

Other Resources

- github, gitlab, etc.
- [more ideas here]

Discussion

Things to do, collectively

- Old mantra:
 - If you are spending resources on something, and so are others, it's a candidate for collaboration
- But also:
 - What things do you not spend resources on, that you wish were better?



EMBEDDED LINUX CONFERENCE



OPEN SOURCE SUMMIT
EUROPE

THE LINUX FOUNDATION

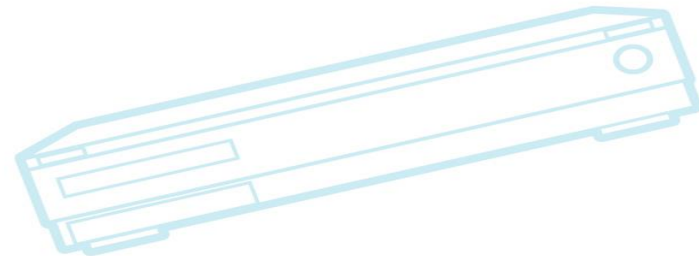
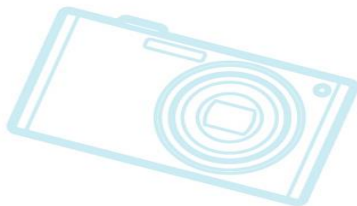
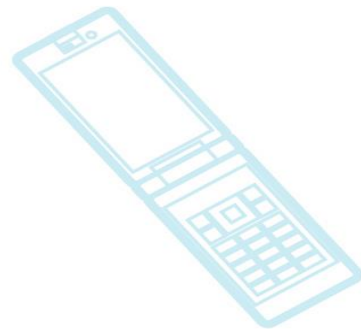




CE Workgroup

Appendix

- Slides to support scorecard assessments





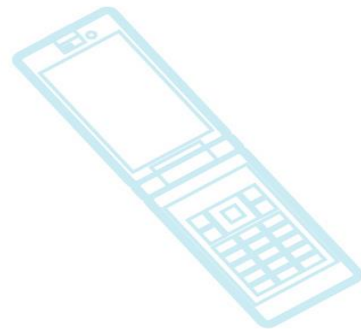
CE Workgroup

Real Technology Scorecard

- System size – done
- Boot time – done
- Power management – done??
- Realtime – done??
- Security – in progress

Recognise now that all of these are “holistic”.

- All of these require pervasive, constant maintenance, distributed throughout the code base





CE Workgroup

Technology Scorecard (reality and explanations)

- System size – **done**
 - Lower limit is about 16M, and it's not getting any better
 - Alas, Linux will never be on 1-cent processors
 - The 10 trillion IoT sensors will be running something else (darn)
- Boot time – **done**
 - Cold boot boot time reduction largely abandoned in mainline
 - People do heroic special-casing to get low boot times, when required
 - Most products use either suspend/resume or low-power idle



CE Workgroup

Technology Scorecard (explanations)

- Power Management – **done??**
 - governors, frequency scaling, power domains, power qos, and power scheduling features are all upstream
 - It requires SoC and board support (e.g. driver pm integration) for it to work
 - It's now mostly a BSP (Board Support Package) problem



CE Workgroup

Technology Scorecard (explanations)

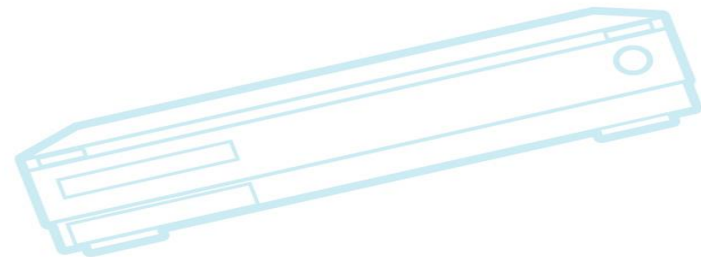
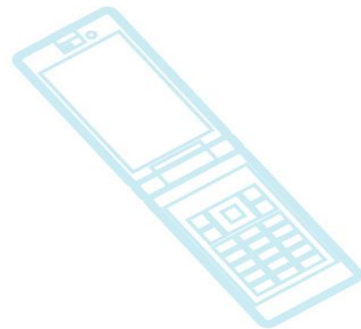
- Realtime – **done??**
 - **PREEMPT_RT** code is (*almost all*) upstream!!
 - But it requires ongoing maintenance to avoid changes that damage realtime performance
- Security – **in progress**
 - kernel hardening, handling security reports, Rust drivers
 - Alpha/Omega project



CE Workgroup

Development Scorecard

- Build systems/Distros
- Training/Consulting
- Toolchains
- Debugging capabilities
- Test Systems
- Hardware support
- options: good, in progress, lacking





Development Scorecard

- Build systems/Distros - **good**
 - Yocto Project, Buildroot, Debian, and specialized ones: OpenWrt, Android
 - Still being worked on, but are mature and featurefull
- Training/Consulting – **good**
 - There have never been more books, tutorials, training
 - Lots of companies and resources available to help build products
 - Pengutronix, Linutronix, Bootlin, Collabora, PathPartner, Mentor, Wind River, Mender, Witekio, Konsulko, Montavista, Timesys, BayLibre
- Toolchains – **good**
 - both gcc and llvm are useful (including for cross-development)
 - SoC vendors (and others) add support for new instructions sets



CE Workgroup

Development Scorecard

- Debugging capabilities – good
 - Lots of options for tracing, debugging, diagnostics
- Test Systems – in progress
 - CI systems and test suites are available
 - But automated test coverage can be improved
 - Automated testing still has gaps (particularly hardware testing)



CE Workgroup

Development Scorecard

- Hardware support – in progress
 - SoC vendors provide support for their chips: Intel, ARM (Linaro), RISC-V, MIPS, etc.
 - Not all of it is upstream
 - Sometimes, it's still hard to get vendors to mainline their hardware support
 - Requires driver work by product makers, when drivers should already be in mainline
 - Product makers carry too much technical debt (ie out-of-tree patches)

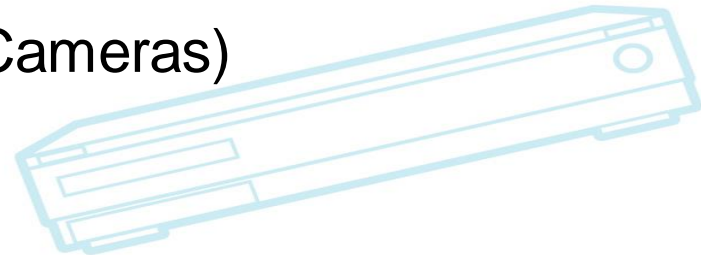
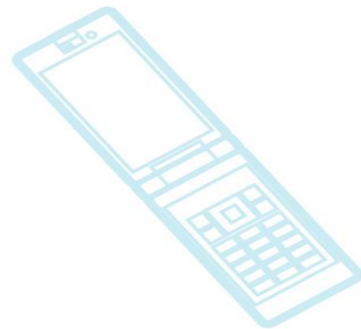


CE Workgroup

Markets Scorecard

- Drones
- Robots
- Cars
- Space systems
- Routers
- Mobile Phones
- Consumer electronics (TVs, DVRs, Cameras)

• *I know this list is not comprehensive*





Verticals Scorecard

- Drones - good
 - Walmart just announced expansion of drone deliveries to 34 states
 - Amazon announced delivery trials in California by end of year
 - Lots of commercial drones running Linux
 - DroneCode project
- Robots – good
 - Robot Operating Systems (ROS) and ROS2
 - Projection that at least 55% of total commercial robots shipped in 2024 will have at least one ROS package
 - See <https://www.businesswire.com/news/home/20190516005135/en/Rise-ROS-55-total-commercial-robots-shipped>
- Cars – good
 - AGL just released UCB 13 in April – instrument cluster and infotainment
 - Tesla self-driving uses Linux
 - Lots of Linux in telematics



Verticals Scorecard

- Space Systems - improving
 - Starlink, SpaceX rockets, Mars helicopter, some cubesats use Linux
 - Commodity hardware in space will probably use Linux more often
- Routers – good
 - Anyone can build a router with Linux these days
- Mobile Phones – good
 - Android market share is 70%
- Consumer electronics (TVs, DVRs, Cameras, Audio) - good
 - In many segments, Linux has almost 100% market share